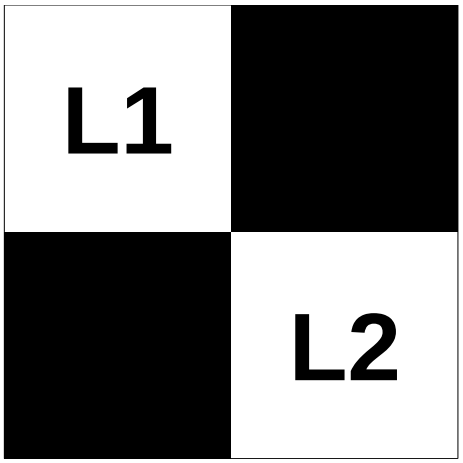
Breaking the Language Barrier:
A Game-Changing Approach



Version 0.22

Ziyuan Yao
yaoziyuan@gmail.com
https://sites.google.com/site/yaoziyuan/

25 March 2012

Table of Contents

Overview

In today's world, the goal of breaking the language barrier is pursued on two fronts: language teachers teaching students a second language, thus enabling humans to manually break the language barrier, and computational linguists building increasingly better machine translation systems to automatically break the language barrier.

However, I see important, unfulfilled opportunities on both fronts:

In second language teaching, amazingly efficient teaching methods have not gone mainstream and not drawn enough attention from computational linguists (so that these methods could be automated and truly powerful). For example, imagine if you're browsing a Web page in your native language, and a Web browser extension automatically detects the topic of this page and inserts relevant foreign language micro-lessons in it, so that you can incidentally learn a foreign language while browsing interesting native language information :-) This AdSense-like "L1-driven L2 teaching" will be the future of second language teaching.

In machine translation, computational linguists only pay attention to computer capabilities to process natural language (known as natural language processing, NLP), and totally ignore human capabilities to share some burden from the computer in language processing, which can lead to significantly better results. For example, theory and practice have proven that syntax disambiguation is a much harder task than word sense disambiguation, and therefore machine translation tends to screw up the word order of the translation result if the language pair has disparate word orders; but what if machine translation preserves the source language's word order in the translation result, and teaches the end user about the source language's word order so that he can manually figure out the logic of the translation result? If the end user is willing to commit some of his own natural intelligence in the man-machine joint effort to break the language barrier, he will get the job done better.

Therefore this ebook presents emerging ideas and implementations in computer-assisted language learning (CALL), second language reading and writing aids and machine translation (MT) that strive to leverage both human and machine language processing potential and capabilities, and will redefine the way people break the language barrier.

Approaches whose titles have an exclamation mark (!) are stirring game-changing technologies which are the driving forces behind this initiative.

You can stay informed of new versions of this ebook by subscribing to

> http://groups.google.com/group/blbgca-announce

and discuss topics in the ebook with the author and other readers at

> http://groups.google.com/group/blbgca-discuss

Chapter 1: Breaking the Language Barrier with Language Learning

Sometimes a person wants to internalize a foreign language in order to understand and generate information in that language, especially in the case of English, which is the de facto lingua franca in this era of globalization.

Section 1.1 “Foreign Language Acquisition” discusses a novel approach to learning a foreign language (exemplified by English).

A person with some foreign language knowledge may still need assistance to better read and write in that language. Therefore, Sections 1.2 “Foreign Language Writing Aids” and 1.3 “Foreign Language Reading Aids” discuss how novel tools can assist a non-native user in writing and reading.

1.1. Foreign Language Acquisition

A language can be divided into two parts: the easy part is its grammar and a few function words, which account for a very small and fixed portion of the language's entire body of knowledge; the hard part is its vast vocabulary, which is constantly growing and changing and can't be exhausted even by a native speaker.

Therefore, the problem of language acquisition is largely the problem of vocabulary acquisition, and a language acquisition solution's overall performance is largely determined by its vocabulary acquisition performance.

The problem of vocabulary acquisition can be divided into two subproblems: “when” – when is potentially the best time to teach the user a word, and “how” – when such a teaching opportunity comes, what is the best way for the user to memorize the word and bond its spelling, pronunciation and meaning all together?

Section 1.1.1 addresses the “when” problem with a method called “L1-driven L2 teaching”, which automatically teaches you a second language when you're browsing native language websites.

Section 1.1.2 addresses the “how” problem with various mnemonic devices, all of them fitting neatly with the “L1-driven L2 teaching” framework.

1.1.1. L1-Driven L2 Teaching! (L1DL2T)

1.1.1.1. The Idea

A Quick Introduction

Imagine if you're browsing a Web page in your native language (“L1”), and a Web browser extension automatically detects the topic of this page and inserts relevant foreign language (“L2”) micro-lessons in it, so that you can incidentally learn a foreign language while browsing interesting native language information :-) For example, if the Web page is a news story about basketball, the browser extension can insert micro-lessons about basketball words and expressions in the foreign language you wish to learn.

This AdSense-like “L1-driven L2 teaching” will be the future of second language teaching.

Topic-Oriented vs. Word-Oriented Teaching

Besides inserting foreign language micro-lessons based on the page's overall topic, the browser extension could even insert micro-lessons after individual words on that page to specifically teach these words' foreign counterparts. For example, if a sentence

> 他是一个好学生。

(Chinese for “He is a good student.”) appears in a Chinese person's Web browser, the browser extension can insert after “学生” a micro-lesson that teaches its English counterpart, “student”:

> 他是一个好学生 (学生是 student)。

(The micro-lesson reads: “STUDENT IS student”.) Additional information such as student's pronunciation can also be inserted. After several micro-lessons like this (each lesson teaching different additional information such as example sentences, related phrases and comparisons to near-synonyms), the computer can directly replace future occurrences of “学生” with “student”:

> 他是一个好 student。

But bear in mind that such direct replacement is not always technically possible or pedagogically welcome, especially if the word being taught is a verb and has different argument structures in the two languages. In practice, a foreign word can be practiced separately in micro-lessons, each lesson containing one example sentence, e.g.

> 他是一个好学生 (学生是 student, 如: *There are 20 students in our class.*)。

Handling Ambiguity in Word-Oriented Teaching

Unlike topic-oriented teaching, word-oriented teaching needs to deal with a problem concerning ambiguous native language words on the Web page. The browser extension needs artificial intelligence (more specifically, “word sense disambiguation”) to determine an ambiguous word's intended meaning based on context, and then teach foreign language for that meaning. Such disambiguation may not be always right, so the computer should **always tell the user which meaning is being assumed in the teaching.**

If an L1DL2T system wants to avoid the problem of word sense disambiguation entirely, it should use topic-oriented teaching instead of word-oriented teaching.

What about Teaching Grammatical Knowledge?

Grammatical knowledge can be taught similarly using L1-driven L2 teaching. The browser extension could detect a certain grammatical usage in the native language page and insert a micro-lesson after that usage to teach its corresponding foreign language grammatical usage.

L1DL2T in a Multi-Peer Environment

If an L1DL2T system inserts foreign language micro-lessons not only into the user's incoming native language communication (e.g. a Web page loaded into his browser) but also outgoing communication (e.g. a message he posts to a forum), all his recipients will be engaged in language learning, even if they themselves do not install an L1DL2T system on their side. Put another way, if only one active participant in an online community (e.g. an IRC chat room or a forum) L1DL2T-izes his outgoing messages, all other members will be learning the foreign language. It's like someone smoking in a lobby – no one else will survive the smoke.

Such a situation also fosters language learners' “productive knowledge” in addition to “receptive knowledge” (“receptive” means a learner can recognize the meaning of a word when he sees or hears that word, while “productive” means he can independently write or say a word when he wants to use it). For example, suppose two Chinese, Alice and Bob, are chatting with each other, and Alice says:

> 他是一个好学生。

(Chinese for “He is a good student.”), but Alice's L1DL2T system transforms this outgoing message, replacing the Chinese word “学生” with its English counterpart, “student”:

> 他是一个好 student。

Now both Alice and Bob see this transformed message, and suppose now Bob wants to say:

> 不，他不是一个好学生。

(Chinese for “No, he is not a good student.”), but he is influenced by the English word “student” in Alice's message, and subconsciously follows suit, typing “student” instead of “学生” in his reply:

> 不，他不是一个好 student。

Thus, Bob is engaged in not only “recognizing” this English word but also “producing” it.

1.1.1.2. Why Is It the Best? A Proof

Why is L1DL2T the best vocabulary acquisition strategy? Below is my proof.

Fact 1: A Cognitive Constraint in Vocabulary Acquisition

The 1999 movie *The Matrix* shows us an advanced brain-computer interface, with which *Neo* is taught kung fu in seconds. Ideally, we would like to have such a tool to impart a foreign language to us in

seconds as well. But this is still science fiction, and we still have to acquire knowledge with natural means: our eyes, ears and slowish memory.

Fact 2: An Affective Constraint in Vocabulary Acquisition

Besides the abovementioned cognitive limit, we humans also have an emotional threshold in acquiring new knowledge. For example, a beginner of a foreign language is not likely willing to sit there all day and memorize a dictionary of this foreign language. Similarly, he would find it very difficult to read an article in this foreign language intended for native readers, as most words in such an article would be unfamiliar to him. These two examples tell us that we simply can't force a person to digest a large number of unfamiliar foreign language words at once.

Divide and Timely Conquer

Based on Fact 2 above, we can conclude that the vast vocabulary of a foreign language must be divided into small “pieces” and taught to a learner one piece at a time (e.g. one word at a time), and we'd better choose a time when the learner is most motivated to learn such a piece. For example:

- when the learner is playing a computer game or participating in a virtual reality world such as *Second Life*, label objects in that game or virtual world with foreign language descriptions;
- when the learner is watching a movie, show both native language and foreign language subtitles on the screen;
- when the learner is using a computer program, show the program's user interface in a foreign language;
- when the learner is interacting with the real world, show foreign language labels on road signs, billboards, products, official documents, etc.;
- when the learner is browsing a native language Web page, insert foreign language micro-lessons relevant to the page's topic or certain words of the page.

Among the above examples, it is obvious that the last example, L1DL2T, provides the widest spectrum of word teaching opportunities.

1.1.1.3. Historical Developments

1960s: Robbins Burling's Diglot Method

In the 1960s American anthropologist and linguist Robbins Burling first introduced the method of gradually introducing foreign language ingredients in a native language storybook to teach foreign language in his seminal paper *“Some Outlandish Proposals for the Teaching of Foreign Languages”*. Burling called it the “diglot method” and was inspired by a *“Learning Chinese”* book series published by Yale University Press, where new Chinese characters gradually replaced Romanized Chinese in a textbook.

Burling's method requires a human translator to manually transform a native language storybook in such a way that, as a reader reads on, he would see more and more words and phrases expressed in a foreign language, and eventually whole sentences and paragraphs too. This forces the reader to look up newly encountered foreign language elements in a dictionary or guess their meanings based on context, therefore gradually picking up that foreign language as he explores the story.

Since then the method is also known as “mixed texts”, “bilingual method”, “code switching”, “sandwich technique” or “diglot weave” in the foreign language teaching research community. However the method has never become a mainstream method.

1990s – Present: Diglot Books, Ebooks and Videos on the Market

There are educational materials using manually prepared diglot texts on the foreign language teaching market, but they have never gone mainstream. For example, *PowerGlide* (www.power-glide.com) sells interactive diglot ebooks; Professors Ji Yuhua (纪玉华) and Xu Qichao (许其潮) sell diglot videos titled *“Three Little Pigs and Stepwise English” (三只小猪进阶英语)* in China.

2000s – Present: Automatic L1DL2T Theories and Systems

In November 2004 I independently came up with the diglot idea (Usenet post and thread), and again in April 2007 (Usenet post and thread). From the beginning I have been researching it as an automatic system (e.g. a Web browser extension) using computational linguistics and natural language processing (CL/NLP). Major aspects of this research are presented in Section 1.1.1.1 “The Idea”. For example:

1. I propose a new L1DL2T paradigm, “topic-oriented teaching”, to completely avoid the problem of word sense disambiguation;
2. I propose that word-oriented teaching must always tell the user which sense is assumed when teaching foreign language for an ambiguous native language word;
3. I propose that we can put L2 teachings and practices in “micro-lessons” separately from a Web page's original L1 text, so that we won't have linguistic problems commonly encountered when two languages are mixed in the same sentence.

I will also present an example L1DL2T system design in Section 1.1.1.4 “An Example System Design”, code name *ATLAS (Active Target Language Acquisition System)*.

There are also other efforts to implement an automatic L1DL2T system in recent years:

- *WebVocab* (http://webvocab.sourceforge.net/) is a kind of Firefox add-on (Greasemonkey user script). It can be classified as “word-oriented teaching” (see “Topic-Oriented vs. Word-Oriented Teaching” in Section 1.1.1.1), and only disambiguates words by part-of-speech clues (e.g. a word after “I” must be a verb/adverb rather than a noun/adjective, so “can” after “I” must be in its auxiliary verb sense, “be able to”, rather than its noun sense, “container”); otherwise it will not teach or practice foreign language for ambiguous words at all.
- *ming-a-ling* (https://addons.mozilla.org/en-US/firefox/addon/ming-a-ling/) is a Firefox extension that can also be classified as “word-oriented” L1DL2T. It simply calls Google Translate to translate native language words on a Web page to a foreign language. However, in case a native language word is ambiguous, it does not tell the user which sense is assumed in the teaching, so the user may be taught a wrong foreign language word and will not know about it when such a misteaching happens.
- *Characterizer* (https://addons.mozilla.org/en-US/firefox/addon/characterizer/) is a Firefox extension that aims to teach Chinese or Japanese characters by putting them in the native language Web page you're browsing. It is also a word-oriented L1DL2T system. However, in case a native language word is ambiguous, it will simply choose a random sense and teach you the Chinese/Japanese character for that sense, and you will never know if a misteaching happens.

- *polyglop* (https://addons.mozilla.org/en-US/firefox/addon/polyglop/) is a Firefox extension that is also a word-oriented L1DL2T. In case a native language word is ambiguous, it will choose a sense at random and teach that sense in foreign language. You will never know if a misteaching happens.
- *polyglot* (https://chrome.google.com/webstore/detail/plpjkjplknknmhfhkjgcfgofclmlnine) is a Chrome extension that also does word-oriented L1DL2T by calling Google Translate to translate certain native language words in your Chrome browser. Like extensions above, it will not tell you which sense is assumed when teaching foreign language for an ambiguous native language word, so you will never know if a misteaching happens.

As you see, all these other efforts listed above are word-oriented L1DL2T systems but they don't tell you which sense is chosen for foreign language teaching when a native language word is ambiguous, so you will never know if a misteaching happens.

I believe an automatic L1DL2T system should either be topic-oriented (so that it doesn't involve the problem of word sense disambiguation at all), or be word-oriented but always tell the user which sense is chosen for teaching.

1.1.1.4. An Example System Design

1.1.1.4.1. Overview

Below I will present the design of a fictional L1DL2T system, code name **ATLAS (Active Target Language Acquisition System)**. It consists of a Chrome extension and associated open standards.

Currently it is only focused on word-oriented teaching. However, the two paradigms (word-oriented teaching and topic-oriented teaching) actually share much in common, and the system has features that will support topic-oriented teaching at a later time. These features will also be presented.

Operation

The Chrome extension will load two files before actual foreign language teaching takes place: (1) an **“ATLAS teaching mission profile”** (or “mission profile” for short) that specifies which native language words will trigger foreign language teaching, what content will be displayed to the user in such a teaching (i.e. the micro-lessons), and data that help the extension to disambiguate ambiguous native language words so that ATLAS can try to teach foreign language for the right sense; (2) an **“ATLAS user profile”** (or “user profile” for short) that records which micro-lessons a user has already been taught.

Therefore the extension's behavior will largely be defined by these two kinds of profiles. Sections 1.1.1.4.2 and 1.1.1.4.3 will present the specifications of these profiles in detail.

Word Sense Disambiguation Strategy

As a word-oriented L1DL2T system, a word sense disambiguation (WSD) strategy is required for

ATLAS. We will use a very simple yet effective WSD approach that assumes a word's intended sense is determined by the “topic” of the word's context, and the “topic” is in turn determined by what other words occur in that context. For example, if the user's native language (L1) is English and there is an ambiguous word “bass” in his Web browser, and if there are words such as “sea” and “fishing” nearby, then the ongoing topic is probably “fishing” and therefore this “bass” is probably in the fish sense; but if there are words such as “music” and “song” nearby, then the ongoing topic is probably “music” and therefore this “bass” is probably in the music sense.

Therefore in an ATLAS teaching mission profile, words will be grouped into “topics” so that they can help disambiguate each other in the same topic. This will be explained in detail in Section 1.1.1.4.2 “ATLAS Mission Profiles”.

Since no WSD strategy is perfect, micro-lessons must explicitly tell the user which word sense is being assumed in the teaching.

Off-Topic Teaching

In principle, word-oriented teaching will teach a foreign language word only if its corresponding native language word appears in the user's Web browser. But what if native language words defined in a teaching mission profile never appear in the browser? For example, what if we have a mission profile that teaches foreign language for basketball words, but the user never happens to browse Web pages related to basketball? In that case the teaching mission would never be performed successfully. To address this problem, we should let ATLAS occasionally teach micro-lessons **unrelated to a Web page's words**, at the bottom of that page, if ATLAS finds it difficult to find normal teaching opportunities.

1.1.1.4.2. ATLAS Mission Profiles

An ATLAS teaching mission profile (“mission profile” for short) is a plain text file with the file extension “.mission.json” that defines what will be taught to the user and how: which native language words will trigger foreign language teaching, what content will be displayed to the user in such a teaching (i.e. the micro-lessons), and data that help the extension to disambiguate ambiguous native language words so that ATLAS can try to teach foreign language for the right sense.

A Sample Mission Profile

I will first show you a sample mission profile “sample.mission.json” and then explain it.

```
{
	"dataFormat": "ATLAS Teaching Mission Profile Format 0.01",

	"title": "School-related words (English -> Chinese)",
	"description": "For English speakers to learn school-related words in Chinese.",

	"authors":
	[
		"Ziyuan Yao (yaoziyuan@gmail.com)"
	],
	"party": "yaoziyuan@gmail.com",
	"date": "21 February 2012",
```

```
      "license": "Creative Commons Attribution 3.0 License",

      "L1": "English",
      "L2": "Simplified Chinese",

      "paradigm": "word-oriented teaching",
      "wotContextWindowSize": {"unit": "word", "value": 30},

      "lexemes":
      [
            {
                  "lexemeID": "teacher1",
                  "inflectedForms": ["teacher", "teachers"],
                  "microLessons": ["A teacher is a 教师 (jiàoshī).", "..."]
            },
            {
                  "lexemeID": "student1",
                  "inflectedForms": ["student", "students"],
                  "microLessons": ["A student is a 学生 (xuéshēng).", "..."]
            },
            {
                  "lexemeID": "blackboard1",
                  "inflectedForms": ["blackboard", "blackboards"],
                  "microLessons": ["A blackboard is a 黑板 (hēibǎn).", "..."]
            },
            {
                  "lexemeID": "teach1",
                  "inflectedForms": ["teach", "teaches", "teaching", "taught"],
                  "microLessons": ["To teach something is to 教 (jiāo) something.",
"..."]
            },
            {
                  "lexemeID": "learn1",
                  "inflectedForms": ["learn", "learns", "learning", "learned", "learnt"],
                  "microLessons": ["To learn something is to 学 (xué) something.", "..."]
            },
            {
                  "lexemeID": "textbook1",
                  "inflectedForms": ["textbook", "textbooks"],
                  "microLessons": ["A textbook is a 课本 (kèběn).", "..."]
            }
      ],

      "topics":
      [
            {
                  "topicTitle": "Schooling",
                  "members":
                  [
                        {"lexemeID": "teacher1", "weight": 10, "dependency": 0},
                        {"lexemeID": "student1", "weight": 10, "dependency": 0},
                        {"lexemeID": "blackboard1", "weight": 10, "dependency": 0},
                        {"lexemeID": "teach1", "weight": 10, "dependency": 0},
                        {"lexemeID": "learn1", "weight": 10, "dependency": 0},
                        {"lexemeID": "textbook1", "weight": 10, "dependency": 0}
                  ]
            },
            {
                  "topicTitle": "...",
                  "members":
                  [
                        {"lexemeID": "...", "weight": ..., "dependency": ...},
                        {"lexemeID": "...", "weight": ..., "dependency": ...},
                        {"lexemeID": "...", "weight": ..., "dependency": ...}
```

```
                    ]
                }
        ]
}
```

The structured format you see in the above sample mission profile is called **JSON** (JavaScript Object Notation). Ideally, mission profiles should be edited with a dedicated editor program, but if you're brave enough to edit it directly, you should learn JSON first, which is very quick to learn.

dataFormat specifies the format version that this mission profile uses.

title, **description**, **authors**, **date** and **license** are descriptive items for human users to read; they don't affect ATLAS's behavior.

party is a feature that allows authors of multiple mission profiles to coordinate their efforts so that their mission profiles can virtually work as a single mission. This will be further explained later. If you don't want to coordinate with other authors, use a unique identity of yours (e.g. email address) as the party value.

L1 and **L2** respectively specify the native and foreign language in this teaching mission. L1 also tells ATLAS whether the native language uses spaces to delimit words. For example, English uses spaces to delimit words but Simplified Chinese doesn't.

paradigm is either “word-oriented teaching” or “topic-oriented teaching”. It tells ATLAS which paradigm to use with this mission profile.

wotContextWindowSize, used with paradigm = “word-oriented teaching”, tells ATLAS's word sense disambiguation (WSD) algorithm how large a context it should look at to determine a word's sense. If L1 is a spaced language (e.g. English), wotContextWindowSize will be specified in terms of words; otherwise (e.g. Simplified Chinese) it will be specified in terms of characters.

totContextWindowSize, used with paradigm = “topic-oriented teaching”, tells ATLAS how large a context it should look at to discover topics that may trigger L2 teaching. If L1 is a spaced language (e.g. English), totContextWindowSize will be specified in terms of words; otherwise (e.g. Simplified Chinese) it will be specified in terms of characters.

lexemes specifies a list of L1 lexemes (actually “lexical items”, as phrases are also allowed) that could trigger L2 teaching. Each lexeme contains (1) a lexemeID that will be unique among mission profiles of the same party (see “party” above) and the same L1, (2) a list of inflectedForms that specify all possible forms that this lexeme may take in a Web page (including the lemma), and (3) a list of microLessons that will take turns to show up each time ATLAS decides to teach foreign language for this lexeme.

topics specifies a list of “topics” that group topically related lexemes together, so that these lexemes can hint at each other in word sense disambiguation (WSD). For example, the “Schooling” topic groups 6 school-related lexemes defined in the “lexemes” section. Each lexeme in a topic is associated with two values: weight and dependency. These values will be further explained later. It is also very important to note that a lexeme can appear in multiple topics, and each appearance will have its own unique set of weight and dependency values. “topicTitle” is a descriptive item for humans to identify a

topic; if paradigm = “topic-oriented teaching”, another item, “topicID”, will be used for the computer to uniquely identify a topic among mission profiles of the same party and L1.

Word Sense Disambiguation with “weight” and “dependency”

The L1 words used in the above sample mission profile (“teacher”, “student”, “blackboard”, etc.) are largely unambiguous, and therefore I can't explain “weight” and “dependency” using that profile. Let's look at another mission profile snippet that involves an ambiguous word, “bass”:

```
	"lexemes":
	[
		{
			"lexemeID": "sea1",
			"inflectedForms": ["sea", "seas"],
			"microLessons": ["A sea is a 大海 (dàhǎi).", "..."]
		},
		{
			"lexemeID": "fishing1",
			"inflectedForms": ["fishing"],
			"microLessons": ["Fishing is 钓鱼 (diàoyú).", "..."]
		},
		{
			"lexemeID": "bass_(fish)",
			"inflectedForms": ["bass", "basses"],
			"microLessons": ["A bass (fish) is a 鲈鱼 (lúyú).", "..."]
		},
		{
			"lexemeID": "music1",
			"inflectedForms": ["music"],
			"microLessons": ["Music is 音乐 (yīnyuè).", "..."]
		},
		{
			"lexemeID": "song1",
			"inflectedForms": ["song”, “songs”],
			"microLessons": ["A song is a 歌曲 (gēqǔ).", "..."]
		},
		{
			"lexemeID": "bass_(music)",
			"inflectedForms": ["bass"],
			"microLessons": ["Bass (music) is 贝斯 (bèisī).", "..."]
		}
	],

	"topics":
	[
		{
			"topicTitle": "Fishing",
			"members":
			[
				{"lexemeID": "sea1", "weight": 10, "dependency": 0},
				{"lexemeID": "fishing1", "weight": 10, "dependency": 0},
				{"lexemeID": "bass_(fish)", "weight": 10, "dependency": 10}
			]
		},
		{
			"topicTitle": "Music",
			"members":
			[
				{"lexemeID": "music1", "weight": 10, "dependency": 0},
				{"lexemeID": "song1", "weight": 10, "dependency": 0},
				{"lexemeID": "bass_(music)", "weight": 10, "dependency": 10}
```

```
                ]
        }
]
```

(Before you read on, bear in mind that it is not necessary for a mission profile author, such as a second language teacher, to really understand “weight” and “dependency”, because she/he can be instructed to always use weight = 10 and dependency = 20 as a default setting.)

As you can see, the word “bass” appears in two topics, “Fishing” and “Music”, respectively as two lexemes, “bass_(fish)” and “bass_(music)”. We want ATLAS to interpret the word “bass” as the lexeme “bass_(fish)”, if either the lexeme “sea1” or “fishing1” appears nearby; or interpret it as “bass_(music)”, if either “music1” or “song1” appears nearby. To do this, we give “sea1” and “fishing1” a “weight” of 10, and give “bass_(fish)” a “dependency” of 10, which means whether “bass_(fish)” will be assumed depends on whether other “Fishing” lexemes (e.g. “sea1” and “fishing1”) in the same context constitute a total weight of at least 10. In other words, either “sea1” or “fishing1” or both of them must appear nearby to let ATLAS interpret the word “bass” as the lexeme “bass_(fish)” and teach this lexeme's L2 micro-lessons. We treat “music1”, “song1” and “bass_(music)” similarly so that at least one among “music1” and “song1” must be present in the context to let ATLAS interpret the word “bass” as “bass_(music)” and teach L2 micro-lessons accordingly.

If a word is unambiguous (e.g. “music”), its lexeme (e.g. “music1”) can have a “dependency” of 0, which means it doesn't require any other lexeme from the same topic to appear nearby to help disambiguate it, as it is unambiguous anyway.

If a lexeme does not suggest a topic as much as other lexemes do, it can have a smaller weight (e.g. weight = 5) for that topic than other lexemes. For example, the occurrence of “ball pen” in a Web page may not suggest the presence of the topic “Schooling” as much as the occurrence of “teacher” would, so “ball pen” may have a smaller weight than “teacher” for that topic. Similarly, there can be lexemes that have a greater weight than others, if they more strongly suggest a topic.

Also note that the micro-lessons for “bass_(fish)” and “bass_(music)” explicitly tell the user which “bass” is being taught in foreign language, by adding “(fish)” and “(music)” in the lesson content. This is in accordance with the “word-oriented teaching must explicitly tell the user which word sense is being assumed when teaching foreign language for an ambiguous native language word” principle I proposed in Section 1.1.1.1 “The Idea”.

“All-for-All” WSD vs. “All-for-One” WSD

Previously we demonstrated “topics” within which lexemes hint at each other in word sense disambiguation (WSD), e.g.

```
        {
                "topicTitle": "Fishing",
                "members":
                [
                        {"lexemeID": "sea1", "weight": 10, "dependency": 0},
                        {"lexemeID": "fishing1", "weight": 10, "dependency": 0},
                        {"lexemeID": "bass_(fish)", "weight": 10, "dependency": 10}
                ]
        }
```

We call this “all-for-all WSD”. However, if necessary, we can even design topics solely for the purpose of hinting at only one lexeme in WSD. For example, we can modify the above topic as:

```
{
	"topicTitle": "Bass (fish)",
	"members":
	[
		{"lexemeID": "sea1", "weight": 10, "dependency": 99999},
		{"lexemeID": "fishing1", "weight": 10, "dependency": 99999},
		{"lexemeID": "bass_(fish)", "weight": 10, "dependency": 10}
	]
}
```

First, notice the topicTitle has been changed to “Bass (fish)”, because this new topic is solely for hinting at the lexeme “bass_(fish)”.

Secondly, notice both “sea1” and “fishing1” have a “dependency” of 99999, which means they'll never be activated for L2 teaching (at least from this topic). But they have a normal weight of 10, which means they can help activate “bass_(fish)” for L2 teaching.

Parties

party is a feature that allows authors of multiple mission profiles to coordinate their efforts so that their mission profiles can virtually work as a single mission. If you don't want to coordinate with other authors, use a unique identity of yours (e.g. email address) as the party value..

If multiple mission profiles specify the same party value, their lexemeIDs are mutually recognized and therefore their lexemes can be merged. Besides, their weight and dependency values are in accordance with the same design standard (e.g. they would all use 10 as a standard weight). Furthermore, they will have consistent wotContextWindowSize and totContextWindowSize values.

Supporting Topic-Oriented Teaching

Although currently this example system design is focused on word-oriented teaching, it will be easy to support topic-oriented teaching because the two paradigms share much data in common.

For example, consider the Schooling topic in our first sample mission profile:

```
{
	"topicTitle": "Schooling",
	"members":
	[
		{"lexemeID": "teacher1", "weight": 10, "dependency": 0},
		{"lexemeID": "student1", "weight": 10, "dependency": 0},
		{"lexemeID": "blackboard1", "weight": 10, "dependency": 0},
		{"lexemeID": "teach1", "weight": 10, "dependency": 0},
		{"lexemeID": "learn1", "weight": 10, "dependency": 0},
		{"lexemeID": "textbook1", "weight": 10, "dependency": 0}
	]
}
```

We can disable word-oriented teaching, by changing all dependency values to a large number, say,

99999. Then we can introduce a topic-level dependency value and microLessons, so that the topic itself can be activated for L2 teaching:

```
            {
                  "topicTitle": "Schooling",
                  "topicID": "schooling",
                  "members":
                  [
                        {"lexemeID": "teacher1", "weight": 10, "dependency": 99999},
                        {"lexemeID": "student1", "weight": 10, "dependency": 99999},
                        {"lexemeID": "blackboard1", "weight": 10, "dependency": 99999},
                        {"lexemeID": "teach1", "weight": 10, "dependency": 99999},
                        {"lexemeID": "learn1", "weight": 10, "dependency": 99999},
                        {"lexemeID": "textbook1", "weight": 10, "dependency": 99999}
                  ],
                  "dependency": 30,
                  "microLessons": ["Lesson 1...", "Lesson 2...", ...]
            }
```

This new topic means if at least three lexemes in this topic are present in a context, ATLAS can insert the topic-level microLessons into that context to teach schooling-related L2 lessons.

“topicID” gives the topic a unique ID which will help ATLAS record the user's learning progress for this topic's micro-lessons.

Note that a mission profile for topic-oriented teaching would usually use a larger context window size (totContextWindowSize) than mission profiles for word-oriented teaching (wotContextWindowSize), because the determination of a large portion of text's topic requires a scan window larger than that of the determination of a word's sense.

Note that a mission profile can choose either word-oriented teaching or topic-oriented teaching (via the “paradigm” item), but not both, because running both paradigms at the same time would involve unnecessary additional engineering complexity to handle conflicts between the two.

1.1.1.4.3. ATLAS User Profiles

An ATLAS user profile (“user profile” for short) is a plain text file with the file extension “.user.json” that specifies a user's identity, preferences and learning records.

A Sample User Profile

I will first show you a sample user profile “sample.user.json” and then explain it.

```
{
      "dataFormat": "ATLAS User Profile Format 0.01",

      "name": "Ziyuan Yao",
      "moreInfo": "Email: yaoziyuan@gmail.com",

      "preferences":
      {
            "learnSameWordFromMultipleParties": false,
```

```
		"microLessonDensity": {"unit": "word", "value": 100}
	},

	"wotLearningRecords":
	[
		{
			"L1": "English",
			"L2": "Simplified Chinese",
			"party": "yaoziyuan@gmail.com",
			"lexemeID": "teacher1",
			"lemma": "teacher",
			"microLessonsCompleted": 1
		},
		{
			...
		}
	],

	"totLearningRecords":
	[
		{
			"L1": "English",
			"L2": "Simplified Chinese",
			"party": "yaoziyuan@gmail.com",
			"topicID": "schooling",
			"microLessonsCompleted": 1
		},
		{
			...
		}
	]
}
```

The structured format you see in the above sample mission profile is called **JSON** (JavaScript Object Notation). Ideally, user profiles will be handled by ATLAS automatically, but if you're brave enough to edit it manually, you should learn JSON first, which is very quick to learn.

dataFormat specifies the format version that this user profile uses.

name and **moreInfo** identify the user.

learnSameWordFromMultipleParties is a boolean value that specifies whether the user is willing to learn lessons about the same word offered by multiple parties. See “party” in Section 1.1.1.4.2 “ATLAS Mission Profiles”. If false, lessons for the same word offered by new parties will be skipped.

microLessonDensity specifies the distance between two adjacent micro-lessons inserted into a Web page. It means how frequently ATLAS should insert micro-lessons.

wotLearningRecords is a list of lexemes whose L2 micro-lessons have been or are being taught to the user via word-oriented teaching.

totLearningRecords is a list of topics whose L2 micro-lessons have been or are being taught to the user via topic-oriented teaching.

1.1.1.4.4. Data Acquisition Strategies

Since ATLAS relies on mission profiles to provide L1-driven L2 teaching in the user's Web browser, it is very important to discuss how such mission profiles can be produced. Generally, I see two approaches:

Let Second Language Teachers Manually Prepare Mission Profiles

Teachers can design lexemes and topics based on:

- what topics their students most likely browse online (e.g. sports, entertainment, fashion, gaming, etc.);
- what topics they're going to teach in the near term;
- general-purpose topics that are likely to appear in any Web page (topics that group related general-purpose words, such as a topic that groups “if”, “then” and “else”, or another topic that groups “ask” and “answer”.

Teachers can also form workgroups that pool their efforts together, using the same **party** value to coordinate their mission profiles (see Section 1.1.1.4.2).

Let Computational Linguists Automatically Generate Mission Profiles

It is even possible for computational linguists to automatically generate ATLAS mission profiles from relevant data sources. For example, a lexeme can correspond to a Wikipedia article, and on that article a computer program can automatically find out links to topically related lexemes (articles), thus automatically generating topics that group related lexemes together. Wikipedia also connects multilingual versions of the same lexeme (article) together, which can be useful for automatically generating L2 micro-lessons for a given lexeme.

1.1.2. Word Mnemonics

The L1-driven L2 teaching (L1DL2T) method discussed in Section 1.1.1 already implies an approach to word memorization: by repetition (a new word is taught and practiced in a series of micro-lessons before it is considered learned). Research into more sophisticated mnemonics has unveiled methods that can serve as powerful force multipliers for L1DL2T, which will be presented in the following sections. Among them, “Phonetically Intuitive English” and “Etymology and Free Association” are recommended by this ebook as **“essential mnemonics”** and I will prove why (Section 1.1.2.1.3).

Phonetically Intuitive English (essential): Memorizing a word in terms of syllables takes far less effort than in terms of letters, and therefore pronunciation as a more compressed form than spelling is a key mnemonic. Section 1.1.2.1.1 “Phonetically Intuitive English” is an approach that integrates a word's pronunciation into its spelling, enforcing correct, confident and firm memorization of pronunciation, which in turn effectively facilitates memorization and recall of spelling.

Etymology and Free Association (essential): Many words are known to be built on smaller meaningful units known as word roots and affixes, or derived from related words. Knowing frequently used roots and affixes and a new word's etymology can certainly help the user memorize the new word

in a logical manner. Even if a word is not *etymologically associated* with any word, root or affix, people can still *freely associate* it with an already known word that is similar in form (either in written form or in spoken form) and, optionally but desirably, related in meaning. Section 1.1.2.1.2 “Etymology and Free Association” revisits these widely known methods.

Orthographically Intuitive English: Certain parts of a long word can be so obscure that they are often ignored even by native speakers, such as a word's choice between “-ance” and “-ence”. Section 1.1.2.2.1 “Orthographically Intuitive English” discusses an approach that deliberately “amplifies” such “weak signals”, so that the learner gets a stronger impression.

Progressive Word Acquisition: Sections 1.1.2.2.2 “Progressive Word Acquisition” is an approach that splits a long word into more digestible parts and eventually conquers the whole word.

Section 1.1.2.3 “Principles Learned” extracts several “principles of word memorization” from the methods discussed in earlier sections, giving us a more fundamental understanding of why these methods work.

1.1.2.1. Essential Mnemonics

1.1.2.1.1. Phonetically Intuitive English! (PIE)

Note: Phonetically Intuitive English is one of two “essential mnemonics” recommended by this ebook, the other one being “Etymology and Free Association” (see Section 1.1.2.1.2). A proof why they are essential is in Section 1.1.2.1.3.

A Quick Introduction

Phonetically Intuitive English slightly decorates or modifies an English word's visual form (usually by adding diacritical marks) to better reflect its pronunciation, while retaining its original spelling. A word can be displayed in this form as it is taught to a non-native learner for the first few times (e.g. by L1-driven L2 teaching; see Section 1.1.1), in order to enforce correct, confident and firm memorization of pronunciation as early as possible, which in turn also facilitates effective memorization of spelling.

A full-fledged PIE sentence may look like this (view with a Unicode font with advanced typography features such as the free and open source “SIL Andika Basic”; in practice, a browser add-on that shows PIE text in a browser will always enforce such fonts for such text to ensure good rendering):

> À qṵic̩k brőw̵n fox jumps ọ̄vèr t͈h̵è lạ̄zy̍ dog.

The above example shows pronunciation in a very verbose mode: “A”, “u”, “c”, “o”, “e”, “t”, “a” and “y” are assigned diacritics to differentiate from their default sound values; “w” and “h” have a short bar which means they're silent; multi-syllable words such as “over” and “lazy” have a dot to indicate stress. Such a mode is intended for a non-native beginner of English, who is unaware of digraphs like “ow”, “er” and “th”.

On the other hand, more advanced learners can use a liter version:

> A qu̯ick brőwn fox jumps ọ̄ver the lą̄zy dog.

Furthermore, words and word parts (e.g. -tion) that a learner is already familiar with also don't need diacritics.

Note that PIE is intended as **a kind of pronunciation guide** that is automatically displayed as the computer teaches new words (as in L1-driven L2 teaching); it is **not intended to be typewritten or handwritten by a student**.

The Chart

See the next page for the chart of Phonetically Intuitive English 2.0 (PIE2), my latest PIE design.

PHONETICALLY INTUITIVE ENGLISH 2.0

GENERAL MARKS (APPLY TO BOTH VOWEL AND CONSONANT LETTERS)

	PIE2 marks	Remarks	Examples
Default values	~	Usually omitted unless necessary. Drawn above vowel letters a, e, i, o and u for short vowels /æ/, /ɛ/, /ɪ/, /ɒ/ (US: /ɑ:/) and /ʌ/. Drawn below a consonant letter for that letter's most typical consonant value; can be drawn above certain consonant letters such as g and p.	(If necessary) bãt, bẽt, bĩt, bõt, bũt; b̰at
Silence	· (above), - (through)	·: A dot drawn above a vowel or consonant letter silences that letter. -: For certain letters such as i, a short horizontal line is drawn through them instead.	takė, pencɨl
Unsupported values	∘	Drawn above a vowel or consonant letter to mean it has a sound value not supported by PIE2, or its value varies depending on context.	o̊ne

VOWEL MARKS (ALWAYS ABOVE VOWEL LETTERS)

	PIE2 marks	Remarks	Examples
Short vowels	~ (default values); ɕ, ɪ, ɔ, ʌ, ◡ (custom values)	~: The "default value" mark (~) can be used when a vowel letter produces its default short vowel value, namely /æ/, /ɛ/, /ɪ/, /ɒ/ (US: /ɑ:/) and /ʌ/ for a, e, i, o and u. ɕ, ɪ, ɔ, ʌ, ◡: In case a vowel letter produces a short vowel other than its default value, a dedicated diacritic will be used to represent each such vowel: ɕ for /ɛ/, ɪ for /ɪ/, ɔ for /ɒ/ (US: /ɑ:/), ʌ for /ʌ/, and ◡ for /ʊ/.	(Default) bãt, bẽt, bĩt, bõt, bũt (Custom) án̆y, privȧte, swȧp, sôn, p̆ŭt
Long vowels	– (letter names)	When – is drawn above a, e, i/y, o or u/w, the letter has a long vowel sound that equals to its name: /eɪ/, /iː/, /aɪ/, /əʊ/ (US: /oʊ/) or /juː/. ū /juː/ also has a weak variant, ŭ /jʊ/.	tāke, ēve, nīce, mōde, cūte; cū̆re
	·· (Middle English-like)	When ·· is drawn above a, i/y, o or u/w, the letter has a "Middle English-like" long vowel sound: /ɑː/, /iː/, /ɔː/ or /uː/. A good mnemonic is that ·· is simply a 90° rotation of /:/, while the IPA letter before /:/ becomes its corresponding Latin letter.	fäther, machïne, cörd, brüte
	~~, ◡◡ (additive cases)	~~: When two ~'s are added above ei/ey or oi/oy, it means /eɪ/ or /ɔɪ/. Note that these two ~'s can be omitted unless necessary. ◡◡: When two ◡'s are added above two adjacent letters such as oo, it means /uː/.	ẽĩght, bõỹ; fo͝od
	\\, // (special cases)	\\: The letter a has a special case where it pronounces the long vowel /ɔː/, and \\ is drawn above a for this case. A good mnemonic is "The word 'fȁll' has falling strokes above." //: The digraphs ou, ow and au can produce a long vowel /aʊ/ which is also not accounted for so far, and // is drawn above o or a for this case. A good mnemonic is "The word 'ő́ut' has outgoing strokes above."	fȁll, ő́ut
Schwa	\	\: When \ is drawn above a vowel letter (a, e, i/y, o, u/w) or r, the letter pronounces /ə/.	fellȁ, ouɹ̀ (UK)
Long schwa	r̈ (as in er, ir, ur, …); two \'s above two letters	r̈: When ·· is drawn above r as in "word", it means this r (along with all adjacent vowel letters) pronounces /ɜː/. Alternatively, two \'s can be drawn above two adjacent letters to collectively represent /ɜː/, e.g. drawn above o and r as in "word".	woɹ̋d (UK); wȍɹ̋d (UK)

CONSONANT MARKS (USUALLY BELOW CONSONANT LETTERS)

	PIE2 marks	Remarks	Examples
Default values	~	Usually omitted unless necessary. Drawn below a letter for that letter's most typical consonant value; can be drawn above certain consonant letters such as g and p.	b̰at; qu̬ick
Secondary values	ɪ	Drawn below or above a consonant letter to represent usually the second most typical value for that consonant letter, e.g. /dʒ/ for d or g, /k/ for c, /ŋ/ for n, /v/ for f, /θ/ for t, /z/ for s, /gz/ for x.	solḍier, c̣lass, siṇg, oḟ, ṭhin, iṣ, exȧmple
Tertiary values	ɪɪ	Drawn below or above a consonant letter to represent usually the third most typical value for that consonant letter, e.g. /ð/ for t, /f/ for g or p (in order to align with g in this case, p has no secondary value), /t/ for d, /z/ for x.	ṯhis, cou̍gh, p̈hone, booke̤d, x̤anadu
R values	r̲; r̀; r̰	~: Drawn below r for /r/. Can be omitted unless necessary. \: Drawn above r for /ə/. Combined with a ~ below r, this will mean /ər/.	r̰ose; ouɹ̀ (UK); ouɻ̰ (US)
/ʧ/, /ʃ/ and /ʒ/	c; ɔ; –	A c, ɔ or – below certain consonant letters (s, c, t, z) denotes /ʧ/, /ʃ/ or /ʒ/. A good mnemonic is that /ʧ/, /ʃ/ and /ʒ/ can correspond to three typical digraphs "ch", "sh" and "zh", and c, ɔ and – resemble the lower left parts of these digraphs (i.e. the bottoms of "c", "s" and "z").	c̣hair, acṭual; ş̣hirt, acţion, maçhine; verş̣ion

STRESS MARKS (BELOW A SYLLABLE'S PRIMARY VOWEL LETTER)

	PIE2 marks	Remarks	Examples
Certain stress	·	Drawn below the stressed syllable's primary vowel letter.	pronunciạtion
Possible stress	··	Drawn below possible stressed syllables' primary vowel letters.	pr̤es̤ent

Why PIE

To learn a new word, two tasks, among others, are involved: learning its pronunciation and its spelling. These two tasks are related, and choosing which to do first makes a lot of difference. If we learn spelling first, we'd be memorizing a (usually long) sequence of letters, which is as tedious as remembering a long telephone number. But if we learn pronunciation first, we'd be memorizing a much shorter sequence of syllables, which can be done in a breeze; then pronunciation can serve as a good catalyst for the subsequent memorization of spelling.

Therefore pronunciation plays a prominent role in word acquisition, and it is worthwhile finding out a good method to learn it.

A big drawback of IPA is that, because it shows pronunciation **separately** from spelling, it gives the user a chance to skip learning pronunciation at all. This is especially the case when the user encounters an unknown word in reading an article: at that moment, the user cares most about the **meaning** of that new word, not the pronunciation, as he doesn't have a need to hear or say that word in real life in the near future. Therefore he is very likely to skip learning the word's true pronunciation in a dictionary, but instead make a **guessed pronunciation** on his own. Making a guessed pronunciation will then lead to two new problems: (a) because the user is a non-native speaker of English, his guessed pronunciation will tend to be error-prone, and therefore he won't dare to commit this guessed pronunciation to his long-term memory very firmly, lest it would be difficult to “upgrade” the guessed pronunciation to the correct pronunciation in the future; (b) the longer the word is, the more uncertainties there are in guessing a pronunciation, making the guess more error-prone, and therefore it's very likely that the user won't dare to guess a complete pronunciation; he would only guess the first two syllables and then jump to the end of the word. For example, I used to memorize “etymology” as just “ety...logy”, “ubiquitous” as just “ubi...ous”, “thesaurus” as just “thes...us”, and so on; this results in both an incomplete, guessed pronunciation and an incomplete spelling in the user's memory.

PIE, on the other hand, eliminates the problems discussed above. Correct pronunciation is made immediately available to the user as he scans through a word's spelling; there is no need to make a “guessed pronunciation” at all. The user will memorize the **correct**, **complete** pronunciation **firmly**, which in turn will facilitate memorization of the **complete spelling**.

Technical Analysis

There are several technical approaches to “adding something above normal text”. You can design a special font that draws letters with diacritics, or Web browser extensions, plugins and server-side scripts that dynamically generate graphics from special codes (e.g. MathML), or HTML “inline tables” as used in an implementation of “Ruby text” (http://en.wikipedia.org/wiki/Ruby_character, http://web.nickshanks.com/stylesheets/ruby.css), or systems that use two Unicode features – “pre-composed characters” (letters that come with diacritics right out of the box) and “combining characters” (special characters that don't stand alone but add diacritics to other characters). The PIE scheme shown above makes use of some combining characters.

In the making of the PIE scheme, I consulted these Wikipedia articles:

- *English spelling* – spelling-to-sound and sound-to-spelling patterns in English (http://en.wikipedia.org/wiki/English_spelling#Spelling_patterns)
- *Combing character* – tables of combining characters in Unicode (http://en.wikipedia.org/wiki/Combining_character)

- *Pronunciation respelling for English* – comparison of respelling schemes in major dictionaries (http://en.wikipedia.org/wiki/Pronunciation_respelling_for_English)

Historical Notes

The general idea of representing a letter's various sound values by additional marks is probably as old as diacritics.

American dictionaries before the 20th century showed diacritical marks directly above headwords to indicate pronunciation for native readers (though not necessarily verbosely). They have been replaced by separate transcription schemes, such as the IPA and respelling systems.

Adding diacritics to English for non-native learners is an obscure method that has never gone mainstream. In China the method was probably first published by George Siao (乔治肖), a retired professor, in the 1970s. He finalized his phonetic transcription scheme in the 1990s based on that of Webster's Dictionary and named it the *"Simple and Easy Phonetic Marks" (简易音标)*. Other people have created more schemes derived from Siao's. Some of them introduced more IPA-like diacritics.

I independently came up with this idea in March 2009 (Usenet post, thread) and created a scheme called *"Phonetically Intuitive English" (PIE)*, based on Unicode. The scheme's chart is shown above. The scheme is so designed that it aims to be the easiest to learn among all existing schemes of its kind.

1.1.2.1.2. Etymology and Free Association

Note: Etymology and Free Association is one of two "essential mnemonics" recommended by this ebook, the other one being "Phonetically Intuitive English" (see Section 1.1.2.1.1). A proof why they are essential is in Section 1.1.2.1.3.

Many words are known to be built on smaller meaningful units known as word roots and affixes, or derived from related words. Knowing frequently used roots and affixes and a new word's etymology can certainly help the user memorize the new word in a logical manner. For example, "memorize" comes from a related word, "memory", and a common suffix, "-ize".

Even if a word is not *etymologically associated* with any word, root or affix, people can still *freely associate* it with an already known word that is similar in form (either in written form or in spoken form) and, optionally but desirably, related in meaning. This already known word can come from the target foreign language, or from the learner's native language. For example, as a Chinese, when I first encountered the word "sonata" in a multimedia encyclopedia as a teenager, I associated it with a traditional Chinese musical instrument *suona* (唢呐) which was featured in an elementary school music class and bears a similar pronunciation to the "sona" part of "sonata". It should also be noted that, as said earlier, sometimes words serving as mnemonics are not necessarily related to the word to be memorized in meaning. For example, to memorize the word "Oscar", we can associate it with two known words, "OS" (operating system) and "car", although they have nothing to do with Oscar in meaning.

Therefore it is useful to let people contribute native language-based and target language-based mnemonics collaboratively online. Wiktionary might be a potential site for such collaboration.

1.1.2.1.3. Why Are They Essential? A Proof

Below I will prove why “Phonetically Intuitive English” and “Etymology and Free Association” are the two and only two “essential mnemonics”, by analogizing word memorization strategies to data compression strategies in computer science.

We will enumerate strategies in data compression, and try to find their counterparts in word memorization:

Compression by removing useless portions: If a portion of a file is useless, we can simply delete it before we compress the file. In the case of word memorization, the same strategy is used by Phonetically Intuitive English (PIE), because PIE lets you memorize a word by pronunciation rather than spelling, and pronunciation doesn't have as many useless phones as spelling when both are read aloud. For example, the word “thesaurus” has a pronunciation that takes 3 syllables: the – sau – rus, but it has a spelling that takes 9 syllables when read aloud: tee – aych – ee – es – ay – you – are – you – es. This is exactly because the latter has many useless phones compared to the former.

Compression by finding redundant portions: If two portions of a file is identical, we only need to represent this identical content in the compression result once. This strategy is also used in word memorization: If we have already memorized “OS” and “car”, we will find it easy to memorize the spelling of “Oscar” because it simply is the combination of two parts that we're already familiar with. Therefore, “Etymology and Free Association” actually uses the same strategy as finding redundant portions in data compression.

Compression by re-coding portions: Data compression has a third strategy which uses shorter codes to encode more frequently occurring portions of a file. This is seen in Huffman coding, for example. In word memorization, this strategy is not used, because re-coding portions of a word when you memorize it would mean manually un-re-coding these portions when you want to reproduce the word, which is prohibitively hard for a human.

Compression by finding the data-generating algorithm: The “algorithmic information complexity” field in computer science says if you can find an algorithm that generates the given data, you don't need to save the data but just the algorithm. Likewise, if we know why a meaning takes a particular word form in a language, we would be able to generate that word form based on the meaning, without memorizing the form itself. Words formed by roots and affixes indeed have some connection between meaning and form, but what about single-syllable words like “cat”, “ant” and “blog”? We can trace *cat*'s etymology back to L.L. *cattus*, which has no earlier etymology and therefore can be considered as randomly formed. We can trace *ant* to W.Gmc. **amaitjo* (**ai-* “off, away” + **mait-* “cut”). Although **amaitjo* has a connection between its form and meaning, this connection gets lost when **amaitjo* further transforms to *ant*, which can be considered as “randomized”. Similarly, *blog* derives from *web log*, a meaningful phrase, but when *web* is shortened to just *b*, the connection between meaning and form is weakened. Therefore, we can not always find a good connection between a word's meaning and form, so we can't just memorize this “connection” to reproduce the word's form.

Therefore we now know “Phonetically Intuitive English” and “Etymology and Free Association” are essential word memorization strategies as the same strategies are used in data compression.

1.1.2.2. Other Mnemonics

1.1.2.2.1. Orthographically Intuitive English (OIE)

PIE in Section 1.1.2.1.1 essentially encodes a word's pronunciation into its spelling. This spawns a symmetric question: can we encode a word's spelling into its pronunciation as well? For example, “reference” and “insurance” have suffixes that sound the same but spell differently (-ence and -ance), and can we slightly modify these suffixes' pronunciations to reflect their spelling difference? Can we give -ance a rising tone and -ence a falling tone? This sounds Chinese and would create new dialects for English that lead to chaos in conversations.

However, if we think outside the box, if we no longer try to “put information into pronunciation”, we may be able to explore other avenues. What about putting this information into a word's visual form? What about lowering the “a” in -ance a little so that it makes a different impression on the learner? So we have

insur_ance

in contrast to

reference

Makes a difference, doesn't it? If the learner develops visual memory that “insurance” has a lowered character in its suffix, then he can infer that this suffix is -ance because -ance has a lowered “a” while -ence doesn't have anything lowered.

We call this “Orthographically Intuitive English” (OIE).

Technical Analysis

Like PIE, OIE makes slight modifications to a word's visual form to add some extra information. Therefore they can share the same techniques for rendering to such effects. In the above example, most document formats that support rich formatting should allow us to raise/lower a character from its baseline. Particularly, in HTML, we can use the tag and its “vertical-align” style property:

```
<p>insur<span style=”vertical-align: -15%”>a</span>nce</p>
```

which will lower the “a” in “insurance” by 15%, making the word look like

insur_ance

Of course, we can also encapsulate the style property into a CSS class so that the above HTML code can shrink to something like

```
<p>insur<span class=”lowered”>a</span>nce</p>
```

Historical Notes

I came up with this idea in August 2009 (Usenet post, thread).

1.1.2.2.2. Progressive Word Acquisition (PWA)

In L1-driven L2 teaching (see Section 1.1.1), long words are optionally split into small segments (usually two syllables long) and taught progressively, and even practiced progressively. This lets the user learn just a little bit each time and pay more attention to each bit (so that he wouldn't just learn an incomplete form of a word as discussed in Section 1.1.2.1.1). For example, when

> 科罗拉多州

(Chinese for “Colorado”) first appears in a Chinese person's Web browser, the computer inserts Colo' after it (optionally with Colo's pronunciation):

> 科罗拉多州 (Colo')

When 科罗拉多州 appears for the second time, the computer may decide to test the user's memory about Colo' so it replaces 科罗拉多州 with

> Colo' (US state)

Note that a hint such as “US state” is necessary in order to differentiate this Colo' from other words beginning with Colo.

For the third occurrence of 科罗拉多州, the computer teaches the full form, Colorado, by inserting it after the Chinese occurrence:

> 科罗拉多州 (Colorado)

At the fourth time, the computer may totally replace 科罗拉多州 with

> Colorado

Not only the foreign language element (Colorado) can emerge gradually, the original native language element (科罗拉多州) can also gradually fade out, either visually or semantically (e.g. 科罗拉多州 → 美国某州 → 地名 → ∅, which means Colorado → US state → place name → ∅). This prevents the learner from suddenly losing the Chinese clue, while also engaging him in active recalls of the occurrence's complete meaning (科罗拉多州) with gradually reduced clues.

1.1.2.3. Principles Learned

This section discusses several “principles of word memorization” learned from word memorization

methods discussed in previous sections, giving us a more fundamental understanding of why these methods work.

Principle of Repetition (used in: L1-Driven L2 Teaching): The more times you learn or use a word, the better you memorize it. This is why L1-driven L2 teaching teaches and practices a new word several times in context before considering it as learned by the user.

Principle of Segmentation (used in: Progressive Word Acquisition): A very long word had better be split into smaller segments and taught gradually. This is the rationale for “Progressive Word Acquisition”.

Principle of Amplification (used in: Orthographically Intuitive English): “Orthographically Intuitive English” deliberately amplifies the difference between similar spellings such as “-ence” and “-ance”, giving the learner a stronger impression and hence better memorization.

Principle of Condensation (used in: Phonetically Intuitive English): You pronounce a word more quickly than you spell it, so pronunciation is a more condense form than spelling and takes much less effort to be memorized. Furthermore, Pronunciation can facilitate the memorization of spelling. So pronunciation should play an early and critical role in word acquisition. “Phonetically Intuitive English” embodies this principle.

Principle of Association (used in: Etymology and Free Association): Memorizing a new word can be made easier if we can reuse already memorized information (words, roots and affixes) to reconstruct it. “Etymology and Free Association” associates a new word with known information etymologically or freely.

Principle of Confidence (used in: Phonetically Intuitive English): We're willing to memorize a piece of information more firmly if we're sure about its long-term validity and correctness, or otherwise we would fear that it would get updated or corrected sooner or later, invalidating what we had already memorized and forcing us to make a great effort to “unlearn” the invalidated version and learn the updated or corrected version anew. Therefore, “Phonetically Intuitive English” prevents us from guessing a new word's pronunciation wrong, and teaches its correct pronunciation from the very beginning.

Principle of Integration (used in: L1-Driven L2 Teaching, Phonetically Intuitive English, Orthographically Intuitive English): Often we are not motivated or self-disciplined enough to learn something, but the computer can “integrate” it into something else that we're highly motivated to engage. For example, we may not want to learn a foreign language word without any context or purpose, but L1-driven L2 teaching can put it in our daily native language reading experience; we may not be very interested in learning a word's pronunciation when it is put separately from the word's spelling, but “Phonetically Intuitive English” can integrate it into the word's spelling; we may not pay attention to the “-ence vs. -ance” difference in a word, but “Orthographically Intuitive English” can reflect this difference information in the word's overall visual shape to which we do pay attention.

1.2. Foreign Language Writing Aids

A person with some foreign language knowledge may still need assistance to better write in that

language. This section discusses how novel tools can assist a non-native user in writing.

1.2.1. Predictive vs. Corrective Writing Aids

In contrast to language learning methods such as L1-driven L2 teaching which builds up the user's foreign language incidentally on a long-term basis, the user also needs just-in-time (“on-demand”) language support that caters into his immediate reading/writing needs. This is especially true of writing, which requires “productive knowledge” that is often ignored in reading, such as a word's correct syntax and applicable context.

On-demand writing aids can be divided into two types:

Predictive writing aids predicts lexical, syntactic and topical information that might be useful in the upcoming writing, based on clues in previous context. Sections 1.2.2 and 1.2.3 discuss two such tools, one for making syntactically valid sentences, the other for choosing topically correct words and larger building blocks such as essay templates.

Corrective writing aids retroactively examines what is just input for possible errors and suggestions. A spell checker is a typical example, which checks for misspellings in input. Corrective writing aids are a much researched area, as most natural language analysis techniques can be applied to examine sentences for invalid usages, and there are studies on non-native writing phenomena such as wrong collocations. Therefore this ebook does not expand this topic.

1.2.2. Input-Driven Syntax Aid! (IDSA)

As a non-native English user inputs a word, e.g. “search”, the word's sentence-making syntaxes are provided by the computer, e.g.

```
v. search:  n. searcher search... [n. search scope] [for n. search target]
```

which means the syntax for the verb “search” normally begins with a noun phrase, the searcher, which is followed by the verb's finite form, then by an optional noun phrase which is the search scope, and then by an optional prepositional phrase stating the search target.

With this information, the user can now write a syntactically valid sentence like

> I'm searching the room for the cat.

Historical Notes

The underlying theories of this idea are widely known in linguistics as case grammar and frame semantics.

1.2.3. Input-Driven Ontology Aid! (IDOA)

As a non-native English user inputs a word, e.g. “badminton”, things (objects) and relations that normally co-exist with the word in the same topic are provided to the user as an “ontology” by the computer, which is a network where there are objects like “racquet”, “shuttlecock” and “playing court”, relations like “serve” and “strike” that connect these objects, and even full-scripted essay templates like “template: a badminton game”.

The user can even “zoom in” at an object or relation to explore the microworld around it (for example, zooming in at “playing court” would lead to a more detailed look at what components a playing court has, e.g. a net) and “zoom out”, just like how we play around in Google Earth.

The benefits of the ontology aid are twofold. First, the ontology helps the user verify that the “seed word”, badminton, is a valid usage in his intended topic; second, the ontology pre-emptively exposes other valid words in this topic to the user, preventing him from using a wrong word, e.g. bat (instead of racquet), from the very beginning.

In case the ontology does not represent the user's intended topic, this means the seed word is wrong. In this case, the computer can guess the user's intended topic based on previous context, show an ontology that represents this topic, and let the user choose a right seed word from this ontology.

Historical Notes

I came up with this idea in March 2006 (Usenet post, thread).

1.3. Foreign Language Reading Aids

Unlike non-native writing, non-native reading doesn't require much help from sophisticated tools. A learner with basic English grammar and the most frequent 100-300 words can engage in serious reading with the help from a point-to-translate dictionary program such as *GoldenDict* and *Babylon* (such a program shows translations for whatever English word or even phrase is under the learner's mouse).

It should be noted that in reading something the learner only cares about the meaning of an unfamiliar word, not further information such as irregular inflected forms. Such further information is taught in L1-driven L2 teaching or timely provided by writing aids, but can also be introduced using the approach below.

A reading aid can insert educational information about a word or sentence into the text being read, just like L1-driven L2 teaching, with the only difference that the main text is in the foreign language rather than the native language. This enables the computer to teach additional knowledge such as idioms and grammatical usages that are beyond word-for-word translation. Word-specific syntaxes as discussed in Section 1.2.2 “Input-Driven Syntax Aid” and domain-specific vocabularies as discussed in Section 1.2.3 “Input-Driven Ontology Aid” are also good feeds.

Chapter 2: Breaking the Language Barrier with Little Learning

Language learning isn't always a cost-effective option to process information in a foreign language, especially if the number of foreign languages involved goes up – an ordinary person certainly doesn't want to acquire the vast vocabularies of the world's many languages, as learning English alone is already demanding. He more likely would like to harness the computer's memory capacity to interpret and generate words in those other foreign languages. Sections 2.1 and 2.2 discuss how the human and the machine can work together to understand and generate information in a foreign language.

2.1. Foreign Language Understanding

How do we understand information in a foreign language, without learning that foreign language? Machine translation (MT) is often the only option. While MT gives us a “gist” about an article's main idea, details are often elusive as MT usually screws up syntax (relations between content words) in the translation result when the language pair has quite different syntactic rules. Therefore, Section 2.1.1 introduces a new approach to MT, where the computer preserves the original language's syntactic structures in the translation result and helps the human reader understand these syntactic structures in their original setting.

2.1.1. Syntax-Preserving Machine Translation! (SPMT)

We will first examine today's machine translation, find out the worst part (syntax disambiguation) that greatly undermines the whole system's usefulness, and then propose a new MT model (“Syntax-Preserving Machine Translation”) that fixes that part.

Today's Machine Translation: Pros and Cons

Before artificial intelligence reaches its fullest potential, machine translation always faces unresolvable ambiguities. The **good news** is, statistical MT such as *Google Translate* disambiguates content words quite well in most cases, and syntactic ambiguity can largely be “transferred” to the target language, without being resolved, if both the source and the target language have common syntactic features. For example, both English and French support prepositional phrases, so

> I passed the test with his help.

can be translated to French without determining whether “with his help” modifies “passed” or “the test” (theoretically, “with his help” can modify “the test” if the test is administered with “his” help). The **bad news** is, syntax disambiguation usually can't be bypassed in a language pair like English to Chinese. In Chinese, “with his help” must be moved to the left of what it modifies, so the Chinese translation result's word order will be either

> I, with his help, passed the test.

or

I passed the with-his-help test.

So, in order to translate the original English sentence to Chinese, it is necessary to determine whether “with his help” modifies “passed” or “the test”.

Inherent AI Complexity in Syntax Disambiguation

Syntax disambiguation like determining what is modified by “with his help” requires capabilities ranging from shallow rules (e.g. “with help” should modify an action rather than an entity, and if there are more than one action – both “pass” and “test” can be considered actions – it should modify the verb – “pass”) to the most sophisticated reasoning based on context or even information external to the text (e.g. in

I saw a cat near a tree and a man.

what is the prepositional object of “near”? “A tree” or “a tree and a man”?)

Let the Human Understand Syntax in Its Original Formation

In the first example in this section, i.e.

I passed the test with his help.

what if “with his help” can be translated to Chinese still in the form of a postponed prepositional phrase, just like how it is translated to French? Then the computer won't have to determine what is modified by “with his help”, as this ambiguity is “transferred” to Chinese just like French.

The Chinese language itself doesn't have postponed prepositional phrases, but we can **teach** a Chinese person what a postponed prepositional phrase is so that we can **introduce** such a prepositional phrase in the Chinese result (the translation will be demonstrated later). To teach language stuff like “what is a prepositional phrase”, L1-driven L2 teaching (see Section 1.1.1) is a good approach. Also, considering syntactic concepts like “preposition” are shared by many languages in the world, it would be quick for a person to learn syntactic knowledge of the world's major languages.

More specifically, we can do machine translation in this way (also see “A Quick Example” below):

- **Content words** are directly machine-translated by a statistical WSD algorithm. In case a content word's default translation doesn't make sense, the user can move the mouse to that translation to see alternative translations.
- **Word order** is generally preserved exactly as it is in the source language text. At the beginning of the translation result, the computer declares the translation result's “sentence word order” (e.g. SVO – subject-verb-object) and “phrase word order” (e.g. head-dependent), so that the user can have a general idea about each sentence and phrase's syntactic structures.
- **Unambiguous or easy-to-disambiguate syntactic features** are automatically translated to *“grammatical markers”*, which are an international standard we'll design to represent syntactic features in a language-independent manner, and is supposed to be learned by the user in

advance. For example, if the computer can positively identify a sentence's subject, it can mark that subject with a “subject marker”, so that the user will know it's a subject. Another example is a verb's transitivity, which can be marked with “vt” or “vi”.

- **Hard-to-disambiguate syntactic features** are left unchanged in the translation result (but may be transcribed to the user's native alphabet for readability). For example, the English preposition “with” is an ambiguous function word, and in case it can't be automatically and confidently disambiguated, we will leave it alone in the translation result, and expect the user to manually learn this word in advance or in place. However, the computer can be certain that “with” is a preposition – this part-of-speech is an unambiguous syntactic feature and the computer can append a “preposition marker” after “with”, to indicate this feature in the translation result. Merely knowing this is a preposition can often enable the user to guess its meaning based on context.

A Quick Example

The computer can translate

> I passed the test with his help.

to Chinese as

> 我 通过了 测试 借助$_{pp}$ 他的 帮助。

which literally means

> I PASSED THE-TEST USING$_{pp}$ HIS HELP.

where the computer translates all content words, preserves the original word order, automatically disambiguates the function word “with” in the “using” sense (as the prepositional object “his help” suggests this sense), and adds a “preposition marker” – “pp” to indicate to the Chinese reader that this is a preposition (so that the reader would realize that it leads a phrase that usually modifies something before it, but not necessarily immediately before it).

Historical Notes

I came up with this idea in March 2010 (Usenet posts 1 and 2).

2.2. Foreign Language Generation

How do we generate information in a foreign language, without learning that foreign language? Machine translation (MT) is often the only option. However, MT doesn't generate publication-quality translation results. Section 2.2.1 introduces an approach to generating text in a foreign language in perfect quality, which requires that the source text be written in unambiguous syntactic structures (well-formed syntax) and content words be disambiguated either automatically or manually.

2.2.1. Formal Language Machine Translation! (FLMT)

A person not knowing a target language can generate information in that language by first writing his information in a formal language – where syntactic structures are written in an unambiguous manner from the very beginning, and content words are from his native vocabulary but will be automatically or manually disambiguated. The formal language composition will then be machine-translated to virtually any foreign language in perfect quality.

A Quick Example

Suppose the user's native language is English. A formal language sentence based on English vocabulary may look like

```
A quick brown fox.jump(over_object: the lazy dog);
```

which literally means “A quick brown fox jumps over the lazy dog”. This formal sentence resembles an object-oriented programming language's function call, where “jump” is a member function of the object “fox”, and “the lazy dog” is the value for an optional argument labeled “over_object”.

Syntactic Well-Formedness

In the process of writing a formal language sentence, an input-driven syntax aid (see Section 1.2.2) helps the user use valid syntax. For example, in writing the above formal sentence, as soon as the user inputs “fox.”, the syntax aid will show actions that a fox can take, and as soon as the user inputs “jump(”, the syntax aid shows possible roles that can be played in a jump event, one of them being something that is jumped over, which is labeled “over_object”.

Lexical Disambiguation

If a content word in such a formal-syntax sentence is ambiguous, automatic word sense disambiguation (WSD) methods can calculate the most likely sense and immediately inform the user of this calculated sense by displaying a synonym below the original word according to this sense. The user can manually reselect a sense if the machine-calculated sense is wrong. All multi-sense content words are initially marked as “unconfirmed” (e.g. using dotted underlines), which means their machine-calculated senses are subject to automatic change if later entered text suggests a better interpretation. An unconfirmed word becomes confirmed when the user corrects the machine-calculated sense of that word, or when the user hits a special key to make all currently unconfirmed words confirmed. This process is like how people input and confirm a Chinese string with a Chinese input method. In addition, if the computer feels certain about a word's disambiguation (e.g. the disambiguation is based on a reliable clue such as a collocation), it can automatically make that word “confirmed” (remove its underline).

Machine Translation to Natural Languages

After all lexical ambiguity is resolved either automatically or manually, the computer can proceed to machine-translating the formal language composition to any target natural language.

Machine Translation to a Standard Form

The *Universal Networking Language (UNL)* Project has been trying to do exactly what is discussed above: machine-translating a formal language composition to all major natural languages, but it hasn't borne fruit for 20 years. This is because natural language generation (NLG) for all major languages is a formidable engineering challenge. An alternative approach I think may work is to machine-translate the formal language composition to a formal but human-readable form like this:

$$\frac{\text{a fox}}{\text{quick, brown}}\ \text{jump over:}\ \frac{\text{the dog}}{\text{lazy}}.$$

Historical Notes

There are quite a few attempts at this approach. The most notable one is the UNL (Universal Networking Language) at http://www.undl.org.

I independently came up with this idea in 2003, by the end of high school.